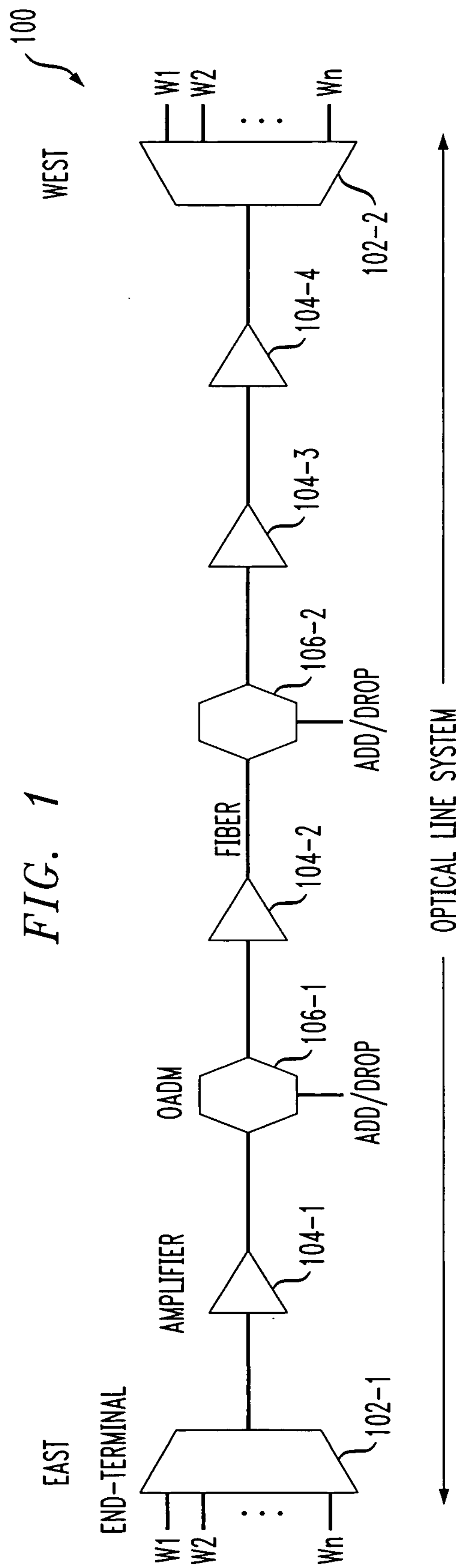
100
WEST
W1
W2
Wn
102-2
104-4
104-3
106-2
ADD/DROP
FIBER
104-2
106-1
OADM
ADD/DROP
AMPLIFIER
104-1
OPTICAL LINE SYSTEM
EAST
END-TERMINAL
102-1
W1
W2
Wn

FIG. 1

FIG. 2

(A)

GENERAL CASE		
PROBLEM	APPROX LOWER BOUND	APPROX UPPER BOUND
$(L, D, *)$	$\Omega(\sqrt{s})$	$O(\sqrt{s})$
(L, U, NE)	$1 + 1/s^2$	2
(L, U, E)	NP-HARD	2
$(C, *, NE)$	IN-APPROXIMABLE	
(C, D, E)	IN-APPROXIMABLE	
(C, U, E)	NP-HARD	$2(1 + \epsilon)$

(B)

SPECIAL CASE	
PROBLEM	COMPLEXITY
$(L, *, E), s = 2$ $\lvert C_2 \rvert = \infty$	POLYNOMIAL
$(L, *, NE), s = 2$	POLYNOMIAL
$(L, U, E), s = 2$	4/3-APPROX
$(L, D, *), s = 3$	NP-HARD

FIG. 3

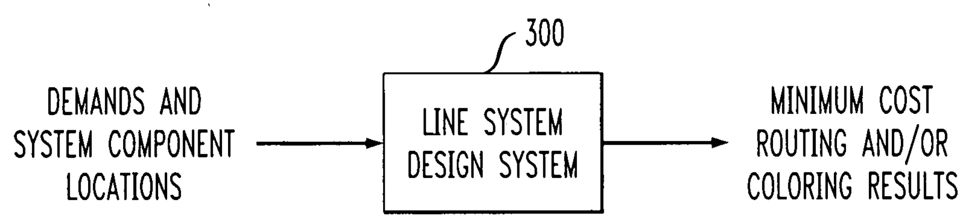

FIG. 4A

```
Methodology A
m(o) = 0
for p = 1 to line system load/2
{
        l(p) = 0; m(p) = m(p-1) + 2;
        for i = 1 to n - 1
        {
                l_i(p) ← load on link e_i
                if(l_i(p) = 0)
                {
                        Divide the line system into two line systems;
                        one from node 0 to node (i-1); the other from
                        node i to node (n-1) and call methodology A
                        on these line systems separately.
                }
                if(l_i(p) > l(p))
                {
                        l(p) = l_i(p)
                }
        }
        create a multigraph G = (V, E), where V = {0, ...n - 1}
        for all demand (i, j) in D
        {
                create an edge (i - 1, j) in G
        }
        for i = 1 to n - 1
        {
                if l_i(p) < l(p)
                        add an edge (i - 1, i) in G
        }
        set the capacity of each edge in G to 1
        find a 2-unit flow from node 0 to node (n - 1) in G
        Let p1 and p2 be the path for the flow
        For all the demands corresponding to links in p1.
        {
                Assign the color c_m(p) to demand
                remove the demand from D
        }
        For all the demands corresponding to links in p2
        {
                Assign the color c_m(p)+1 to demand
                remove the demand from D
        }
}
```

FIG. 4B

```
Routing Phase:
    if (L(R_s) ≥ n(1 + ε)/ε)
        Output R_s
    else {

        Compute D_1 = {d ∈ D| d in any routing goes through at least n/3 links}
        Compute D_2 = D - D_1
        Compute R_1 = the set of all possible routings for demands in D_1
        Compute R_2 = the set of all possible routings for demands in D_2
        in which at most 3S demands are not routed on shortest paths
        Compute R = R_1 x R_2
        Compute r ∈ R such that L(r) = min_{r'∈R} L(r')
        Output r

    }

Coloring Phase:
        U = D
        M = the set of available colors
        l = min_{e_i∈L} l_i(U) (the min. load of demands in U)
        while (l > 0) {
                Compute 0 = H(U) (see below)
                Compute m = {i, j| i, j are the smallest two colors in M }
                Color demands in 0 with colors in m
                U = U - 0
                M = M - m
                l = min_{e_i∈L} l_i(U)
        }
                if (U ≠ Ø) {
                        Color U using methodology A

"Compute 0 = H(U)":
    Compute d_0 = a demand in U that goes through the largest number of links in L
    0 = {d_0}
    L' = set of links covered by demands in 0
    i = 1
    while (L' ≠ L) {
        Compute D_i = {d| d∈U - 0 & d overlaps with d_{i-1}}
        Compute d_i = {d| d∈D_i & d goes through the largest number of links in L - L'}
        i = i+1
    output 0
    }
```

FIG. 4C

```
Methodology B:
e_0 = (-1, 0)
e_{n+1} = (n, n + 1)
L = L∪{e_0, e_{n+1}}
D = D∪{(0, 0), (n + 1, n + 1)}
for all (0 ≤ i ≤ j ≤ n + 1) {
    P(i, j) = Ø
    R(i, j) = Ø
    best = 0
    for all (i ≤ i′ ≤ j′ ≤ j) {
        E_1 = {e_i, e_{i+1}, ... e_{i′}} ∪ {e_{j′+1}, e_{j′+2}, ... e_j}
        E_2 = e_{i′+1}, e_{i′+2}, ... e_{j′}}
        Compute coloring C using methodology b1 where E_1 (E_2) links are colored
        with 1 (2) steps
        if(C ≠Ø) {
            if(i′ - i + j - j′ + 1 ≥ best) {
                R(i, j) = C
                best = i′ - i + j - j′ + 1
            }
        }
    }

}
Compute L_1 = {e_i| e_i ∈ L, l_i ≤ |C_1|}
for all (e_i, e_j ∈ L_1) {
    Compute D_{i,j} = {d| d∈D, d goes through either link e_i, e_j}
    Compute P_{i,j} = coloring obtained by coloring the interval graph D_{i,j} with colors in C_1
}
for all (e_i, e_j ∈ L_1, i < j {
    best = 0
    for all(m, i < m < j) {
        Compute the coloring K = P(i, m) + P(m, j)
        If(K=Ø) continue
        Compute n = number of links that are in one step in K
        if(best < n) {
            best = n
            C = K
        }
    }
    Compute n = number of links that are in one step in R(i, j)
    if(best < n) {
        best = n
        C = R(i, j)
    }
    P(i, j) = C
}
Output P(0, n + 1)
```

FIG. 4D

Methodology b1:
Compute C = interval graph coloring of demands D_1 using colors in C_1
if(C == $\emptyset$) **Output** C
Compute C' = interval graph coloring of the demands in $D - D_1$ using first available colors
Output $C' \cup C$

FIG. 4E

Methodology c1:
$V = \{0, 1, \ldots, n-1\}$
$E = \emptyset$
for all demands $((i, j) \in D - D_1)$ {
    $E = E \cup \{(i - 1, j)\}$
    Directed link $(i - 1, j)$ has unit capacity
}
for all links $(e_i \in L)$ {
    $E = E \cup \{(i - 1, i)\}$
    Directed link $(i - 1, i)$ has capacity $|C_1| + |C_2| - l_i$
}
Graph $G = (V, E)$
Compute maxFlow = Max. Flow f in G from node 0 to node $n - 1$
if(maxFlow < $|C_2|$) **Output** $\emptyset$
Compute $F_1 = \{d | f$ puts zero flow on the edge $(i - 1, j)$ where demand $d = (i, j)\}$
Compute $F_1 = F_1 \cup D_1$
Compute K_1 = coloring that colors demands in F_1 with colors in C_1 only using interval graph coloring
Compute K_2 = coloring that colors demands in $D - F_1$ with colors in C_2 only using interval graph coloring
Output $K = K_1 \cup K_2$

FIG. 4F

METHODOLOGY *D*:

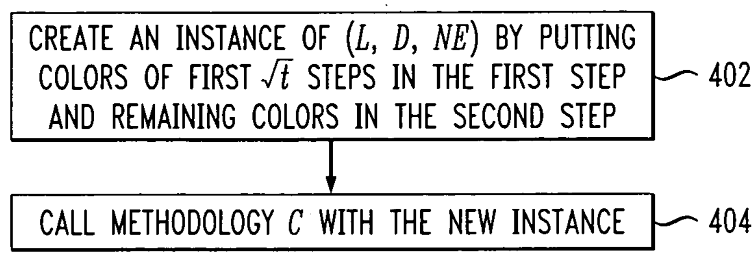

FIG. 4G

METHODOLOGY *E*:

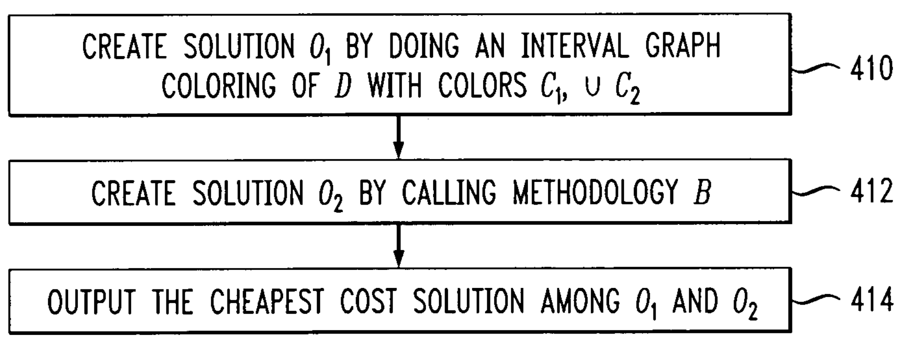

FIG. 5